Image based Shape Approximation by a Genetic Algorithm

Raffael Theiler*
Bsc. Neuroinformatics, University of Zurich

Abstract

Humans are able to understand shapes intuitively. This is not the case for computer systems, where generalized data is preferred. Given a natural or artificial shape, my work proposes ideas on how to find a generalized approximation using an genetic algorithm based on a model derived from the superellipse.

1 Introduction

Reducing complexity helps to pass information to others and sketches are one way to achieve that. How simple it is for us humans to perform this task is fascinating. One may ask the question if there is a computer program that is able to produce an abstract output from an real world image as well.

The first experiment to achieve that included a model of Bezier curves fitted to an image by a genetic algorithm. This lead to the following insights:

- It is not clear beforehand how complex the best approximation is, meaning how many lines are required to preserve the core information of an input. This is contradicting with the fixed genome size of the regular genetic algorithm. Sometimes, using one single line would converge towards a better solution than using multiple. Figure 1 for example shows a mountain view where many lower order curves would not converge towards an identifiable image.
- Finding a fitness function to describe what a "fit" sketch is is unfeasible because information in its core is hardly generalizable. A consistent score may not be accurate because humans would value differently by understanding the scenes context.

Figure 1: *Output from the first algorithm iteration: An approximation of a mountain scenery by one higher order* $(n = 30)$ *Bezier curve. However, this or similar solutions were not reliably reproducible.*

It was ultimately clear that a restriction of the problem is necessary. This paper is focusing on the approximation of preselected, extracted shapes, possibly coming form such a complex scenery, which could be one of the later building blocks to the original task. A combination of different strategies may solve the original problem.

*e-mail:raffael.theiler@uzh.ch

2 Problem Description

It was shown by [Gielis 2003] that many natural and geometrical forms can be described when polar coordinates are introduced to the set of super ellipses as in eq. (4). The name superformula (SF) was choosen by Gielis. The huge variety of shapes that can be calculated with the fixed set of parameters and the relatively low computational effort make the SF feasible for a genetic approach. The concept for this algorithm shall be that an already extracted shape is provided and then, the software finds a close SF approximation to the input.

3 Interpretation of the Input Data

The notation $\boldsymbol{A}$ (bold uppercase) is used for matrices, $\boldsymbol{a}$ (bold lowercase) for vectors.

An image in the form of eq. (1) where positive directions are aligned with the euclidean x and y coordinates is given as input.

$$\boldsymbol{I}(x,y) = \begin{bmatrix} a_{m1} & & \dots & a_{mn} \\ \vdots & \ddots & & \\ a_{11} & a_{21} & \dots & a_{1n} \end{bmatrix} \qquad (1)$$

$$x=\{1,2,\dots,n-1,n\}, \quad y=\{1,2,\dots,m-1,m\}, \quad a\in[0,1]$$

This image is first convolved (operator $*$) with an gaussian filter $G(x,y,r,\sigma)$, r is the radius of the matrix and σ the standard deviation to allow local improvements towards the line. The convoluted matrix

$$\boldsymbol{C}(x,y) = \boldsymbol{G}(x,y,\sigma) * \boldsymbol{I}(x,y) \qquad (2)$$

is then used for the optimization process. Figure 2 shows an example input. The gaussian filter parameters were subject to overall algorithm optimization.

Figure 2: *Two inputs convolved with* $\boldsymbol{G}$*. For large* r, σ *the algorithm did converge faster towards a less optimal solution. Corner regions however were neglected. For small values, the chance of early convergence towards a wrong optimum was increased.*

3.1 Symmetry

SF curves may be shapes with rotational, reflective symmetry or no symmetry at all. A SF curve may as well express a logarithmic spiral [Gielis 2003]. To lower the degree of flexibility, a given input was mirrored by inverting the y axis. The intention of this was to prevent the algorithm from finding solutions that only partially represent the given structure body which may be the case for many different sets of parameters. This would lead to a behavior where the software would never find the same solution twice.

4 Fitness Function

The superelipse, also known as Lamé curve, was originally introduced by Gabriel Lamé in 1818:

$$\left[\frac{x}{a}\right]^n + \left[\frac{y}{b}\right]^n = 1 \tag{3}$$

Equation (3) was used by Gielis almost 200 years later [Gielis 2003] to derive the superformula:

$$r(\varphi) = \left(\left| \frac{\cos\left(\frac{m_1\varphi}{4}\right)}{a} \right|^{n_2} + \left| \frac{\sin\left(\frac{m_2\varphi}{4}\right)}{b} \right|^{n_3} \right)^{-\frac{1}{n_1}}. \tag{4}$$

which is in this application tansformed to euclidian coordinates:

$$x(\varphi) = r(\varphi) \cdot \cos(\varphi)$$
$$y(\varphi) = r(\varphi) \cdot \sin(\varphi)$$

Given an input image $\boldsymbol{C}$, this approach introduces two terms to calculate its fitness. The first operator (F_p) is a score of how precise that the points are aligned with the input curve. To calculate F_p, the Gielis function values $\boldsymbol{p}$ are calculated and transformed to the image domain d_x (given by the function $d_x = dim(\boldsymbol{C}, 1)$):

$$\boldsymbol{p}(s) = \begin{bmatrix} \{x(\varphi_1), y(\varphi_1)\} \\ \vdots \\ \{x(\varphi_n), y(\varphi_n)\} \end{bmatrix}, \varphi_k = \frac{n \cdot 2\Pi}{s}, n = \{1, 2, \ldots, s-1, s\}, \tag{5}$$

The transformation is then applied to each tuple $p \in \boldsymbol{p}$ using aggregated values from $\boldsymbol{p}$ calculated as: $max_x = Max(\boldsymbol{p}_x)$ and $min_x = Min(\boldsymbol{p}_x)$

$$p = \left\{ \frac{(x - min_x) \cdot d_x}{max_x - min_x}, \frac{y \cdot d_x}{max_x - min_x} \right\} \tag{6}$$

The part wise function in limits the output curve to the image. inImage(a, b) is a boolean function that returns $true$ when the pixel is inside the image region.

$$in(a, b) = \begin{cases} \boldsymbol{C}(a, b) & \text{inImage}(a, b) \\ 0 & \neg\text{inImage}(a, b) \end{cases} \tag{7}$$

Finally, F_p is the sum of partial scores. $\lfloor x \rceil$ is used as the nearest integer function:

$$F_p(\boldsymbol{p}, s) = \sum_{n=1}^{s} in(\lfloor \boldsymbol{p}_{n,x} \rceil, \lfloor \boldsymbol{p}_{n,y} \rceil), \quad F_p \in \mathbb{R}_{\geq 0} \tag{8}$$

The second operator (F_v) discards curves that are collapsing at one specific point. For that, the distance in between discrete Gielis points is calculated as vector $\boldsymbol{d}$:

$$\boldsymbol{d}(s) = \begin{bmatrix} \sqrt{|\boldsymbol{p}_{1,x} - \boldsymbol{p}_{2,x}|^2 + |\boldsymbol{p}_{1,y} - \boldsymbol{p}_{2,y}|^2} \\ \vdots \\ \sqrt{|\boldsymbol{p}_{n-1,x} - \boldsymbol{p}_{n,x}|^2 + |\boldsymbol{p}_{n-1,y} - \boldsymbol{p}_{n,y}|^2} \end{bmatrix}_{n=\{2,3,\ldots,s-1,s\}} \tag{9}$$

A shifted sigmoid function is used to map values of $\boldsymbol{d}$ to $[0, 1]$

$$S(x) = \frac{1}{1 + e^{-20(x-0.5)}} \tag{10}$$

F_v uses $Q(p, n)$ which is the n-quantile value of a set p.

$$F_v(\boldsymbol{d}, n) = S(Q(\boldsymbol{d}, n)), \quad F_v \in [0, 1] \tag{11}$$

The final fitness score is then calculated by multiplication.

$$F(\boldsymbol{d}, \boldsymbol{p}, s, n) = F_v(\boldsymbol{d}, n) \cdot F_p(\boldsymbol{p}, s), \quad F \in \mathbb{R}_{\geq 0} \tag{12}$$

5 Algorithm Design

5.1 Genotype and Phenotype

The genotype is a binary representation of the 6 parameters $a, b, m1, m2, n1, n2, n3$ of eq. (4). Each parameter range is limited by an initialization array to a domain preventing division by zero and similar issues. The binary number resolution is globally defined by a constraint because the parameters are value wise close to each other.

The encoding is related to the algorithms behavior when undergoing mutation. The encoding influence was looked at in detail. Given binary encoding, the distance in between a mutated genotype and its original version varies largely depending on the position. Therefore leading to the consideration to use a less sensitive encoding which maintains a fixed Hamilton distance in between two adjacent nodes. However it was shown that there is little difference in between the two procedures [Chakraborty and Janikow 2003].

5.2 Parent Selection

Parents were selected with proportionally increasing probability judged by their fitness scores. This method is known as the "wheel of fortune". A possibility for an elite process is adjustable in the program but was set to a low value in practice because it showed that, while speeding up the convergence process, it does reduce the possibility to converge towards an optimal solution.

5.3 Genetic Operators

Mutation was applied with a given probability to the bits of a genotype. The crossing function uses a double crossing approach with one randomly selected parent. A segment in between a random lower and upper boundary of the first parents genotype is exchanged with the other.

6 Results

Figure 3 shows different shapes approximated by the algorithm. One of the main issues is that the algorithm sometimes converges towards a less optimal solution. Attempts were made to prevent this by limiting elite breeding and increasing the mutation rate to promote population diversity. The parent to child ratio was lowered as well. However after $\sim$ 200 iterations the general direction was determined most of the time.

Initially the structure would sometimes collapse at one point of the input curve, having nearly optimal score with all points being very close to each other. This was solved by introducing the distance related optimization term. It is designed to strongly separate good from bad solutions while maintaining single-objective optimization. Otherwise the two scores would have formed a Pareto optimal front where the selection of solutions would need to be evaluated by SPEA2 or a similar algorithm. [Zitzler et al. 2002].

7 Known Issues and Limitations

7.1 Fixed Angles for SF Discretization

Discretization is applied to the SF by selecting discrete angle values in the polar domain. This leads to the undesired side effect that points in the euclidean space are not uniformly distributed along the function trajectory. For small changes of the function value $r(\varphi)$ points are close to each other. This overvalues areas where the

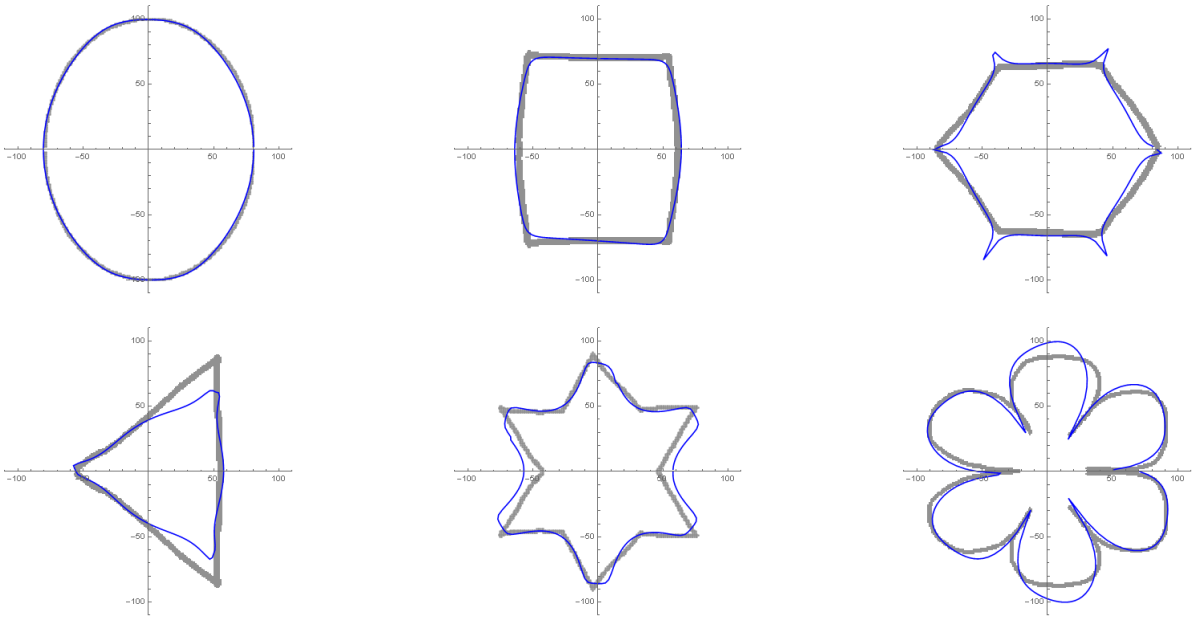

Figure 3: *Hand drawn shapes (gray) were selected with increasing difficulty and approximated by the algorithm (blue). The program was manually terminated after* $\sim$ 600 *steps. (mutation rate: 0.1, crossing rate: 0.5, no elite process. 30% of the population was replaced after each step)*

function is moving slow. As a consequence, areas in fast moving parts are undervalued. Figure 4 shows an example of a "rocket" function where the solution converged towards the nose.

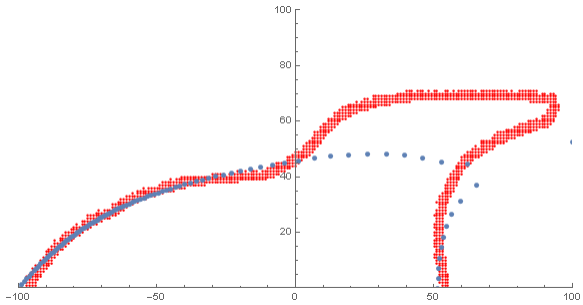

Figure 4: *The algorithm converged towards a solution (x-axis symetric) that has most of its points in quadrant 1 (and 3). Solutions that would better fit the desired curve in quadrant 2 (and 4) count less towards the total fitness than an optimization in quadrant 1. Red points represent the input, blue the discrete SF values*

7.2 Partially Convex vs Part. Concave Solutions

It is possible to converge towards solutions for the same problem where the lobes are either concave or convex. The algorithm was not able to evolve in between the two solutions because the distance in the parameter space is large and both solutions are close to an optimal solution as shown in fig. 5.

7.3 Scaling Issues

Scaling was originally introduced to allow solutions that fit the problem up to a fixed scale $d_x/(max_x - min_x)$. However, this only works if the input image is also realigned with the boundaries. This error was overlooked at initially. As a solution, the input was scaled to the x bounding box as well. This solution has the restriction that curves are not able to overshoot by a small margin in this domain while they can in the y domain.

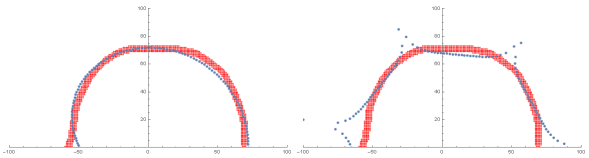

Figure 5: *Two example solutions of a circle approximation. After evolving towards one of the solutions, the algorithm could not switch to the other, more optimal solution.*

7.4 Vanishing Corners

Sometimes, an appropriate amount of corners was selected in the early stage of convergence but then later on lost in favor of a shape with a higher score but less corners. This was observable in regions with flat angles such as the hexagon. While the shape with a lower degree of corners was certainly more accurate in terms of its position, it was less recognizable for humans. Optimization of the filter matrix (σ, r) lowered the probability of this scenario.

8 Related Work

Gielis himself worked on a genetic algorithm that approximates curves [Fougerolle et al. 2013]. Outside of genetic algorithms, the Levenberg-Marquardt method may be used to approximate SF curves [Gavin 2017]. SF curves were successfully applied to detect road signs [Valentine Vega 2012].

References

CHAKRABORTY, U. K., AND JANIKOW, C. Z. 2003. An analysis of gray versus binary encoding in genetic search. *Information Sciences 156*, 3, 253 – 269. Evolutionary Computation.

FOUGEROLLE, Y. D., GIELIS, J., AND TRUCHETET, F. 2013. A robust evolutionary algorithm for the recovery of rational gielis curves. *Pattern Recognition 46*, 8, 2078 – 2091.

GAVIN, H. P. 2017. The levenberg-marquardt method for nonlinear least squares curve-fitting problems.

GIELIS, J. 2003. A generic geometric transformation that unifies a wide range of natural and abstract shapes. *American Journal of Botany 90*, 3, 333–338.

STOOP, R., 2016. Theorie und simulation neuronaler netze. University Lecture.

VALENTINE VEGA, DÉSIRÉ SIDIBÉ, Y. F. 2012. Road signs detection and reconstruction using gielis curves. *International Conference on Computer Vision Theory and Applications*, 393–396.

ZITZLER, E., LAUMANNS, M., AND THIELE, L. 2002. Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In *Evolutionary Methods for Design, Optimisation, and Control*, CIMNE, Barcelona, Spain, 95–100.